**Bootstrap**

**Los 3 pilares de Bootstrap**

**GRID:** Nos va a permitir definir cómo se van a estructurar y adaptar los distintos elementos de nuestra página Web.

**COMPONENTES:** Elementos HTML reutilizables que ya vienen con un estilo predeterminado.

**ÍCONOS:** Incluye iconos para utilizar en los proyectos.

**La Grid de Bootstrap**

Es como una cuadricula que nos va a permitir definir como queremos que se presentar y alinenar el contenido de una página Web.

En Bootstrap se usan cuadriculas, que nos va a permitir trabajar en filas y columnas, pero cada fila va a estar dividida en **12 columnas.**

CLASES PARA LA GRID

.row : si agregamos la clase .row estamos diciéndole que ese elemento va a actuar como una fila. Ej: <div class=”row”></div>

.col- :

.col-xs-: extra small = <576px

.col-sm- : small = >= 576px

.col-md- : medium = >= 768 px

.col-lg- : large = >= 992px

.col-xl- : extra large = >= 1200px

.col-xxl- : extra exctra large = >= 1400px

**CONTENEDORES**

Cada contenedor puede tener **FILAS** y cada fila puede tener hasta **12 COLUMNAS**

<!--Es importante poner el nombre de la class que pertenezca a Bootstrap-->

<div class="container">

    <div class="row">

        <div class="col"></div>

        <div class="col"></div>

    </div>

    <div class="row">

        <div class="col"></div>

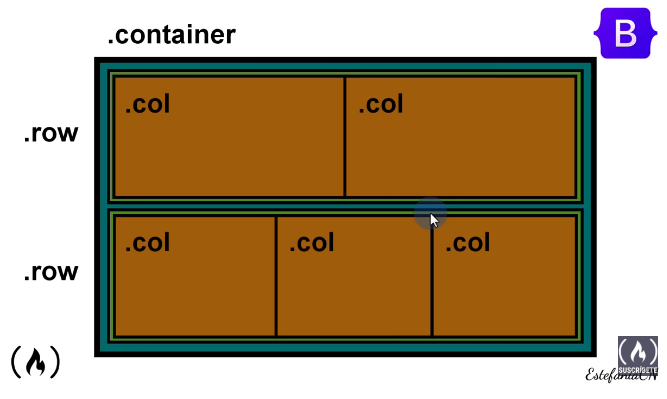
        <div class="col"></div>

        <div class="col"></div>

    </div>

</div>

Se vería así:



**2 TIPOS DE CONTENEDORES EN BOOTSTRAP**

**.container:** crea un contenedor responsive con un ancho máximo fijo que depende del tamaño del dispositivo.

**.container-fluid:** crea un contenedor responsive que cubre el 100% del ancho de la ventana.

**PARA AGREGAR BOOTSTRAP A NUESTROS PROYECTOS**

Vamos a la pagina getbootstrap.com -> Docs -> Downloads -> CDN via js Delivr -> Copiamos el link y el script y lo agregamos al index.html (con esto hace que la pagina descargue lo que necesita de bootstrap cuando se carga, y no hace falta que lo instalemos de manera local en el proyecto)

Ej:

<!DOCTYPE html>

<html lang="es">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"

        integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"

        integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"

        crossorigin="anonymous"></script>

    <title>Document</title>

    <style>

        div {

            height: 250px;

            background-color: purple;

            color: white;

            font-size: 50px;

            margin: 20px;

        }

    </style>

</head>

<body>

    <div class="container">

        .container

    </div>

    <div class="container-fluid">

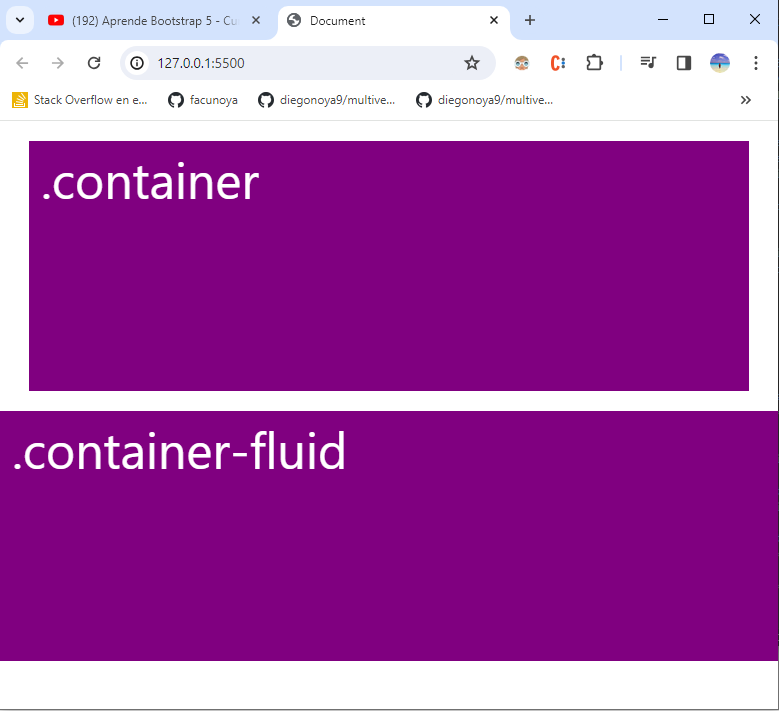
        .container-fluid

    </div>

</body>

</html>

Quedaría así:

****

**Los CONTENEDORES RESPONSIVE:**

**.container-sm**

**.container-md**

**.container-lg**

**.container-xl**

**.container-xxl**

<div class="container text-center">

text-center es una propiedad de bootstrap que nos permite centrar texto desde su clase.

**COMPONENTES**

Un componente en bootstrap es un elemento HTML reutilizable que ya viene con un estilo predeterminado y que podemos usar en nuestra página web. También podemos personalizar el estilo de los componentes de bootstrap.

**FLEXBOX**

Nos permite que los elementos responsive ubicados dentro de un contenedor se distribuyan automáticamente en base al tamaño del dispositivo.

<div class="d-flex">

    Mi contenedor Flex

</div>

d-flex indica que va a ser un contenedor que usa flexbox

flex-direction : Establece el eje principal del contenedor, la dirección en la cual se van a colocar los elementos dentro del contendor. Puede terner estos valores.

row

row-reverse

column

column-reverse

justify-content : como se distribuyen los elementos en el eje principal depende de los 4 anteriores.

Align-items : difine como se distribuyen los elementos en el eje secundario.

**\*METADATOS\***

Un metadato es la codificación de los caracteres que se indica con el atributo charset=”utf-8”

  <meta charset="UTF-8">

    <meta name="author" content="Facundo Noya">

    <meta name="description" content="Portfolio de Facundo Noya">

    <meta name="keywords" content="HTML, CSS, Javascript, React">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

**charset**: indica la codificación de los caracteres

**name** y **content** podemos asignar valores a esos nombres, el viewport es el único que siempre va así: width=device-width, initial-scale=1.0 esto nos va a permitir que se ajuste el tamaño de la ventana del navegador al tamaño del dispositivo.

**Agregar un favicon a la pagina**

<link rel="icon" type="image/x-icon" href="img/Fanku.png">

**rel** y **type** van siempre igual.

En ese ejemplo probé con la imagen que uso en GitHub y la tomo por ser .png, averiguar un poco más del tema, ya que me gustaría modificarla para que sea redonda y de momento no sé cómo se modifica un **link** del **head.**

**Escribí tu respuesta: …**

**HACER IMÁGENES REDONDAS**

class="rounded-circle"

es como un border-radius: 50%;